# Java Project On Employee Management System Using Java As Front End And MySQL As Backend

**Name:-** T. TULASI VISHNU VARDHAN SAI

**Regno:-** 192225081

**Java code :-**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

public class ES {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/company";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "Vardhan@123";

public static void main(String[] args) {

try {

Connection connection = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\n1. Add Employee");

System.out.println("2. View Employees");

System.out.println("3. Update Employee");

System.out.println("4. Delete Employee");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline character

switch (choice) {

case 1:

addEmployee(connection, scanner);

break;

case 2:

viewEmployees(connection);

break;

case 3:

updateEmployee(connection, scanner);

break;

case 4:

deleteEmployee(connection, scanner);

break;

case 5:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please try again.");

}

} while (choice != 5);

scanner.close();

connection.close();

} catch (SQLException e) {

System.out.println("Error: " + e.getMessage());

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

private static void addEmployee(Connection connection, Scanner scanner) {

try {

System.out.print("Enter employee name: ");

String employeeName = scanner.nextLine();

System.out.print("Enter employee department: ");

String employeeDepartment = scanner.nextLine();

System.out.print("Enter employee salary: ");

double employeeSalary = scanner.nextDouble();

String insertQuery = "INSERT INTO employees (name, department, salary) VALUES (?, ?, ?)";

PreparedStatement insertStatement = connection.prepareStatement(insertQuery);

insertStatement.setString(1, employeeName);

insertStatement.setString(2, employeeDepartment);

insertStatement.setDouble(3, employeeSalary);

int rowsInserted = insertStatement.executeUpdate();

if (rowsInserted > 0) {

System.out.println("New employee added.");

} else {

System.out.println("No new employee added.");

}

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

private static void viewEmployees(Connection connection) {

try {

String selectQuery = "SELECT \* FROM employees";

Statement selectStatement = connection.createStatement();

ResultSet resultSet = selectStatement.executeQuery(selectQuery);

System.out.println("\nEmployee ID\tEmployee Name\tDepartment\tSalary");

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String department = resultSet.getString("department");

double salary = resultSet.getDouble("salary");

System.out.println(id + "\t\t" + name + "\t\t" + department + "\t\t" + salary);

}

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

private static void updateEmployee(Connection connection, Scanner scanner) {

try {

System.out.print("Enter employee ID to update: ");

int employeeId = scanner.nextInt();

System.out.print("Enter new salary: ");

double newSalary = scanner.nextDouble();

String updateQuery = "UPDATE employees SET salary = ? WHERE id = ?";

PreparedStatement updateStatement = connection.prepareStatement(updateQuery);

updateStatement.setDouble(1, newSalary);

updateStatement.setInt(2, employeeId);

int rowsUpdated = updateStatement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("Employee updated.");

} else {

System.out.println("No employee updated.");

}

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

private static void deleteEmployee(Connection connection, Scanner scanner) {

try {

System.out.print("Enter employee ID to delete: ");

int employeeId = scanner.nextInt();

String deleteQuery = "DELETE FROM employees WHERE id = ?";

PreparedStatement deleteStatement = connection.prepareStatement(deleteQuery);

deleteStatement.setInt(1, employeeId);

int rowsDeleted = deleteStatement.executeUpdate();

if (rowsDeleted > 0) {

System.out.println("Employee deleted.");

} else {

System.out.println("No employee deleted.");

}

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

}

}

**Output:-**

